Special issue on International Workshop on Network Sensing System

Trans. of the Society of Instrument and Control Engineers
Vol.E-S-1, No.1, 43/51 (2006)

SHR: Stateless Hierarchical Routing for Dynamic Sensor Networks[†]

Niwat THEPVILOJANAPONG *, Yoshito TOBE ** and Kaoru SEZAKI ***

In this paper, we present *Stateless Hierarchical Routing* (SHR), a new routing protocol for multi-hop, wireless sensor networks. The design of the protocol is based on the requirements of sensor networks that every sensor node periodically transmits sensed data to the base station. Tree construction is initiated by the base station which will broadcast a control packet to discover child nodes. Sensor node receiving this packet decides an appropriate parent node to which it will attach, it then broadcasts the same control packet to discover child nodes in the next level of the tree. Consequently, the hierarchical tree is rapidly created without flooding of any control packets. By knowing only a parent address, each node can make forwarding decisions regardless of the knowledge on other neighbors or geographical location. When comparing to other proactive routing protocols, SHR avoids periodic updating for routing maintenance but it can agilely recovers from link failures by switching to a new parent. We evaluate the performance of SHR by using the *ns*-2 simulator and comparing its performance with that of both DSR and AODV. The simulation results demonstrate that SHR has much higher delivery ratio and lower delay on various situations, both static and dynamic networks.

Key Words: wireless sensor networks, routing protocol, hierarchical tree, performance evaluation, simulation

1. Introduction

Recent advances in MEMS-based sensor technology and low-power RF design have enabled the development of relatively inexpensive and low-power wireless sensors $^{1),2)}$. A great number of such sensors can coordinate amongst themselves to achieve a larger sensing task both in urban environments and in inhospitable terrain. They can be used in various applications such as environmental and habitat monitoring, tracking system, failure detection, intrusion detection $^{3)\sim 8)}$. Sensor network needs to be structured differently from traditional mobile ad hoc networks (MANETs) due to its specific communication pattern.

To motivate the challenges in designing a routing protocol, we show a scenario usually happens in any sensing applications. A large number of sensors (over one thousand sensors, for example) are deployed in a remote terrain. These sensors coordinate to establish a communication network, monitor specified tasks, and report sensed data periodically or spontaneously to the base station. When the existing sensors are out of order due to numerous reasons, they reorganize by themselves to repair failed routes. The user may deploy additional sensors to mitigate a severe effect of many failed nodes, thereby enforcing the sensors to reconstruct in order to take advantage of the added system resources. Hence, we consider a routing protocol based on a specific communication pattern which is also robust to dynamic natures of sensor networks.

The remainder of the paper is organized as follows. Section 2 enumerates the detailed mechanisms of our routing protocol. Section 3 evaluates the performance of proposed protocol in simulated networks through *ns*-2 simulation tool. Section 4 describes related work and we summarize our work in Section 5.

2. Stateless Hierarchical Routing Protocol

We design a *Stateless Hierarchical Routing* (SHR) protocol for large-scaled, dynamic sensor networks. First, we describe a specific pattern of communication required in wireless sensor networks as well as dynamic natures of sensor networks. Since SHR is based on hierarchical tree, we then present how the tree is hierarchically constructed and how the sensor nodes adapt to dynamic networks. The details of SHR protocol are as follows.

2.1 Objectives and Requirements

We targets on various applications, e.g., environmental

[†] Presented at INSS2004 (2004.6)

[*] Department of Information and Communication Engineering, University of Tokyo, 7-3-1 Hongo Bunkyo Tokyo 113-8656, Japan. E-mail: wat@mcl.iis.u-tokyo.ac.jp

[**] Department of Information Systems and Multimedia Design, Tokyo Denki University, 2-2 Kanda-Nishikicho Chiyoda Tokyo 101-8457, Japan. E-mail: yoshito@unl.im.dendai.ac.jp

[***] Center for Spatial Information Science (CSIS), University of Tokyo, 4-6-1 Komaba Meguro Tokyo 153-8505, Japan. E-mail: sezaki@iis.u-tokyo.ac.jp

(Received October 31, 2004)
(Revised April 14, 2005)

and habitat monitoring, tracking system, failure detection, intrusion detection $^{3)\sim 8)}$. One example of ongoing work is habitat monitoring on the Great Duck Island $^{5)}$. In such applications, tasks of all sensor nodes are sensing specified values and reporting sensed data to the sink. Therefore, we need a protocol to do many-to-one communications originated from a number of sensor nodes, and destined at a sink node. Our target applications described above only require this kind of communication.

The design of SHR has been driven by the following goals and requirements.

- **Simplicity and Scalability**. Since unconstrained scale is an inherent feature of a sensor network and the sensors have limited computing capability as well as memory resources, we seek to minimize the number of operations performed and the states maintained at each sensor. In particular, each node does not maintain all neighboring nodes and the path calculation is not based on the complex algorithms such as Dijkstra's or Bellman-Ford algorithm $^{9)}$.
- **Robustness**. Our solution provides self-organized mechanisms in order to deal with the dynamic natures of sensor networks, i.e., joining and leaving scenarios.

2.2 Network Model

We consider a sensor networks composed of a small number of base stations or sinks and a numerous number of wireless sensors randomly distributed in an interesting area. These sensor nodes have limited processing power, storage, bandwidth, and energy, while the base stations have powerful resources to collect and process sensor readings. We assume that the sensor nodes are not mobile nodes, i.e., all nodes are fixed for the duration of their lifetime, however, sensor network we consider has dynamic characteristic such that new nodes may be deployed at any time or the battery of the node is depleted with time. In particular, the sensor nodes have omni-directional antennas and use RF to communicate. All wireless network transmissions are inherently broadcast.

We design a routing protocol for sensor networks whose communication pattern differs from conventional mobile ad hoc networks. Let N be a set of all nodes in the network except the base station (BS). Previous works $^{10)\sim 15)}$ for a set of communicating parties (s, d), where $s \in \{N, BS\}$ and $d \in \{N, BS\}$, while our work is a routing protocol for multipoint-to-point communication, where $s \in \{N\}$ and $d \in \{BS\}$. Namely, every sensor node tries to report sensed data to the base station.

2.3 Tree Construction

SHR is based on hierarchical tree where a base station is a root node, and the sensor nodes are the internal or leaf nodes of the tree. The base station initiates the tree construction by broadcasting (1) two *child request* (`CREQ`) packets separated with the interval T_i. Using two broadcast packets increases reliability of the protocol because broadcasted packet is prone to lose and no any retransmission mechanism supports. *Nonmember node*, a node which does not attach to the tree yet, determines its parent from received `CREQ` packets by choosing a node whose `CREQ` packet has arrived first as a parent or waiting for T_{creq} seconds in order to collect a number of candidates in a *candidate list* and choose a node whose defined metric is the best one (highest received power strength, highest remaining energy, for example). The node has a choice to maintain or delete the candidate list after choosing the parent. It then sends a *child reply* (`CREP`) packet to the selected parent so as to inform that it will be a child node or a leaf node of the current tree. *Member node* which is an internal or leaf node drops the `CREQ` packet immediately. In our implementation, nonmember node waits for a short period of time (T_{creq}) to collect the candidate parents and choose a node whose power strength is the highest and more than a threshold in order to avoid the problem of communication gray zones reported by Lundgren et al. $^{16)}$.

Upon receiving the `CREP` packet, the parent node notifies an acceptance of new child node by replying with a *child acceptance* (`CACP`) packet. The child node waits the `CACP` packet for a period of T_{cacp} seconds and if the `CACP` packet does not arrive within this period, it sends the second `CREP` packet. If the T_{cacp} period has passed again and the `CACP` packet still does not arrive, it sends the third `CREP` packet as a last reply and chooses a new parent for the next round of the T_{cacp} period. After receiving the `CACP` packet from the parent, the child node does the same process as its parent by broadcasting a `CREQ` packet to discover its own children. These procedures are performed by every node in the network. An example of a hierarchical tree created by SHR is shown in **Fig. 1**.

2.4 Joining Mechanism

Joining in sensor networks means the user deploys new sensors into the current network. A newly deployed sensor or nonmember node must find a parent for communicating purpose by broadcasting a *parent request* (`PREQ`) packet. Any member nodes of the tree that hear this packet reply by unicasting a `CREQ` packet to the joining node. Note

---

(1) *Broadcasting* means the transmission of a packet from a source node to every node within its radio coverage. We use this definition throughout the paper.

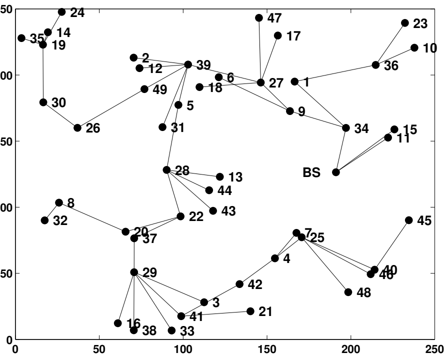


Fig. 1 Hierarchical tree created by SHR protocol. There are 50 nodes in 250m by 250m square region.

that this CREQ packet is not broadcasted as described in Section 2. 3, therefore only one packet is sufficient. Then, the processes will follow the tree construction phase, i.e., the joining node sends a CREP packet to the selected parent and waits for a CACP packet before any communication can begin. We can also limit the number of children per parent in order to distribute the loads and reduce congestion. If the joining node does not receive any CREQ packet, it can infer that no any node is within its radio coverage or all of its neighboring nodes do not attach to the tree yet. In this case, it waits for an incoming CREQ packet after one of its neighbors has attached to the tree.

2. 5 Leaving Mechanism

Leaving means a member node loses communication with other members due to numerous reasons. For instance, the battery of the node is depleted with the time, the node can be damaged due to harsh environment or by the enemy. Before explaining the processes in reconstructing the tree, we give some ideas on how to detect a left node. As a common characteristic of sensor networks, the sensor periodically transmits its sensed data to the base station. When combining this characteristic with our hierarchical protocol, if a parent node does not receive the packets from any children for a while, it infers that such children have completely left from the network. Another method can be done by relying on the underlying MAC layer protocol. We use the latter method (MAC based approach) which is less complex to detect the left nodes in our implementation.

If there is no any acknowledgement on MAC layer after sending the data packet to the parent, a node infers that its parent has left from the network due to any reasons. It immediately switches to a new parent by choosing the most appropriate one from the candidate list (if any) and sending a CREP packet to the selected parent. If there is no any parent in the list or the candidate list is not available, it broadcasts a PREQ packet as if it is a newly deployed node. However, its child nodes will not reply to this PREQ packet to prevent routing loop. Every node that attaches to the orphaned node does nothing because they do not know the absence of their grandparent. They can still forward the packets to the orphaned node as usual, and the orphaned node keeps received packets in its buffer for sending later. In the worst case that the orphaned node does not have any parent in the list and no any response to the PREQ packet, it broadcasts a *parent query* (PQRY) packet to its child nodes asking whether they have the candidate for the parent node. The child nodes reply with a *parent reply* (PREP) packet containing such information. Then, the orphaned node randomly chooses a child node that has at least one candidate parent as its new parent by sending the CREP packet to inform a new relation, and that child node will switch to a new parent chosen from the list. If all of its child nodes do not have any candidate parent, the orphaned node randomly chooses one child node as a new parent by sending the CREP packet as usual and let this selected child node find a new parent by using the PREQ packet. Note that the last scenario is very rare case that may occur in sparse network.

2. 6 Data Communication and Discussions

A great advantage of SHR is that it relies only on the knowledge of a parent node. Therefore, the state required at each node is negligible, and independent of network density and network size which means that SHR is very scalable. In particular, each node just forwards its sensed data and all of received packets to its parent. Thereby, SHR is nearly stateless, i.e., only one parent address suffices for routing purpose. We note again that the candidate list is an option. It is a trade-off between quickness of recovery and freshness of information. If the nodes always broadcast the PREQ packet to discover a new parent without relying on the candidate list, it will get fresh information but it must wait for a reply from the neighboring nodes. Routing table and geographical information are also not necessary. Moreover, route discovery does not use flooding, thereby no propagation of routing information or packets throughout the network. Furthermore, SHR does not apply periodic updating that reduces traffic load so much.

Since a main cause of energy consumption in sensor networks is communication cost compared to computational cost (transmitting a single bit of data is equivalent to 800 instructions [17)]), we can decrease an energy consumption by minimizing the number of transmissions. Instead of forwarding the packets immediately, each node waits for a short period of time to *append* or *aggregate* the data from other nodes and sends them together in one packet. Let us assume T_f denotes a basic forwarding period and F_d denote a delay factor, where $F_d \geq 0$. A node forwards data every $(T_f \times F_d)$ forwarding period. If $F_d = 0$, the node forwards data as soon as it has new sensing or receive a new data. However, if the amounts of data fill up all vacant spaces of one packet, the node will immediately forward the packet regardless of the forwarding period. Aggregation 18), 19) which is a summarization of data is more complex than appending and it depends on applications. Data compression 19) can also be done to reduce the packet size as well as network loads.

The protocol described in this section is summarized as pseudo-codes in Algorithm I through V (**Figs. 2–6**). Algorithm I shows the main procedure of SHR protocol while the others are the functions called by the main algorithm. The *flag_prt* is a flag indicating the existence of parent node. This flag is initially set to DOWN (Algorithm I, line 2) which means that the node does not have a parent yet and it will change the status to UP when the nonmember node has attached to the tree (Algorithm III). The *num_crep* is the number of the `CREP` packets the node sent. This variable is used to determine timeout of the selected parent and it is initially set to 0 (Algorithm I, line 3). When the sensor node is deployed in the field, it immediately broadcasts the `PREQ` packet (Algorithm I, line 7) and wait for an incoming packet as described in Section 2. 4. The incoming packet composes of both data and routing packet. The node follows line 9 (Algorithm I) in the case of data packets and it conforms line 17 (Algorithm I) for the routing packets.

```
Algorithm I:
void main() {
    flag_prt ← DOWN // status of parent: UP, DOWN.
    num_crep ← 0 // number of CREP packets sent.
    flag_choose_prt ← NO // status of calling choose_parent().
    flag_crq_sent ← NO // status of CREQ packet sent.
    BS broadcasts CREQ packet
    sensor node broadcasts PREQ packet
    while rcv_pkt do // rcv_pkt is a received packet.
        if rcv_pkt is data packet then
            if destination is base station then
                if flag_prt = UP then
                    forward rcv_pkt to parent
                else // flag_prt = DOWN
                    buffer rcv_pkt in a queue
                end if
            end if
        else // rcv_pkt is routing packet.
            if rcv_pkt is CREQ packet then
                rcv_creq() // Algorithm II
            else if rcv_pkt is CREP packet then
                if (flag_prt = UP) || (my_addr = BS) then
                    send CACP packet
                end if
            else if rcv_pkt is CACP packet then
                rcv_cacp() // Algorithm III
            else if rcv_pkt is PREQ packet then
                if (my_addr = BS) || ((flag_prt = UP) &
                    (ID_src ≠ parent)) then
                    unicast CREQ packet to a communicating
                        party
                end if
            else if rcv_pkt is PRQY packet then
                send PREP packet containing necessary
                    information
            else if rcv_pkt is PREP packet then
                randomly choose a new parent from its
                    children according to received information
                send CREP packet to inform a new relation
            end if
        end if
    end while
}
```

Fig. 2 The main algorithm of SHR protocol.

3. Performance Evaluation

To evaluate the performance of SHR, we use the *ns*-2 20) simulation tool to run a number of simulations described in this section. We compare the performance with two well-known ad hoc routing protocols, Dynamic Source Routing (DSR) 11) protocol and Ad-Hoc On-Demand Distance Vector (AODV) 12) protocol, which have been shown to offer higher packet delivery ratio than other ad hoc routing protocols 21).

3. 1 Methodology and Metrics

The *ns*-2 simulator includes full simulation of the IEEE 802.11 physical and MAC layers. Our simulations use this MAC layer and assume symmetric links. Using this MAC layer does not affect the evaluation because we need to evaluate the network layer of three protocols. We randomly placed 50 sensor nodes in a 250m by 250m square region. Each node has fixed radio coverage of 50 meters. Note that the nodes have fixed positions without any movement for the entire simulation. We use constant bit rate (CBR) as our traffic sources. A 64-byte data packet is used for all CBR sources. The transmission rates of CBR sources are 0.25, 0.5, 1, and 2 packets per second, i.e., 128, 256, 512, and 1024 bps, respectively, while the bandwidth of sensor nodes is set to 19.2 kbps (2). The

---

(2) Mica2 has bandwidth of 38.4 kbaud encoded with manchester code 22).

```
Algorithm II:
void rcv_creq() {
    if ($flag\_prt$ = UP) || ($my\_addr$ = BS) then
        drop CREQ packet
    else
        $candidate\_list \leftarrow (ID_{src}, metric)$
        if $flag\_choose\_prt$ = NO then
            $flag\_choose\_prt \leftarrow$ YES
            call $choose\_parent()$ at $T_{creq}$ seconds later
                // Algorithm 5
        end if
    end if
}
```

Fig. 3 The *rcv_creq*() function used in SHR protocol.

```
Algorithm III:
void rcv_cacp() {
    $num\_crep \leftarrow 0$ // reset this variable for future parent
        selection.
    send all buffered packets in the queue to $parent$
    if ($flag\_prt$ = DOWN) & ($flag\_crq\_sent$ = NO) then
        broadcast CREQ packet
        $flag\_crq\_sent \leftarrow$ YES
    end if
    $flag\_prt \leftarrow$ UP
}
```

Fig. 4 The *rcv_cacp*() function used in SHR protocol.

```
Algorithm IV:
void choose_parent() {
    if BS is in $candidate\_list$ then
        $parent \leftarrow$ BS
    else
        for all members in $candidate\_list$ do
            $parent \leftarrow$ node whose $metric$ is the best
        end for
    end if
    send CREP packet to $parent$
    $num\_crep \leftarrow num\_crep + 1$ // This variable is used as a
        timeout.
    call $wait\_cacp()$ if CACP packet does not arrive within $T_{cacp}$
        seconds // Algorithm V
}
```

Fig. 5 The *choose_parent*() function used in SHR protocol.

CBR agent will be attached to a UDP agent, which in turn attached to the source node. For all simulations, the communication patterns are peer-to-peer and the starting time of connections is randomly selected. One node from each simulation is randomly chosen as a base station and it is only one destination for all traffic sources, while other 49 nodes are the source nodes (one flow per one source). Each simulation is last for 200 seconds.

```
Algorithm V:
void wait_cacp() {
    if $num\_crep > 2$ then // the node does not receive CACP
        packet from selected parent after sending 3 CREP packets.
        $candidate\_list \leftarrow candidate\_list - parent$ // remove
            current parent.
        $parent \leftarrow$ NULL
        $num\_crep \leftarrow 0$ // reset this variable for future parent
            selection.
        if $|candidate\_list| > 0$ then // there is at least one
            candidate in the list, the node chooses a new parent.
            immediately.
            $choose\_parent()$ // Algorithm IV
        else // The node does not have any candidate.
            periodically broadcast PREQ packet until getting
                CREQ packet
        end if
    else // retransmit CREP packet.
        $num\_crep \leftarrow num\_crep + 1$
        send CREP packet again
        call $wait\_cacp()$ if CACP packet does not arrive within
            $T_{cacp}$ seconds
    end if
}
```

Fig. 6 The *wait_cacp*() function used in SHR protocol.

To compare between various protocols, we choose to evaluate them according to the following two metrics.

- *Data packet delivery ratio*: the ratio between the number of data packets received by the destination and the number of data packets sent by the source.
- *Average delay*: the average one-way latency observed between transmitting a data packet and receiving it at the destination.

The parameters of our protocol used in the simulations are set as follows: $T_i = 0.1$ second (the interval between two `CREQ` packets), $T_{creq} = 0.1$ second (the time waiting for collecting candidate parents), and $T_{cacp} = 0.3$ second (the time waiting for the `CACP` packet from selected parent).

SHR is a proactive routing protocol, while both of DSR and AODV are reactive protocol. We try to achieve the fair comparisons in the simulations for such opposite approaches. In general, SHR launches the tree construction at the beginning of the simulation which is an advantage of proactive approach. In contrast, both reactive protocols discover the route when there is a packet destined for a new destination. To make the opposite approaches begin to discover the route at the same time, every traffic source is forced to issue data packet at the beginning of the simulations (3). However, both DSR and AODV show

---

(3) This is a feasible issue in sensor networks when a num-

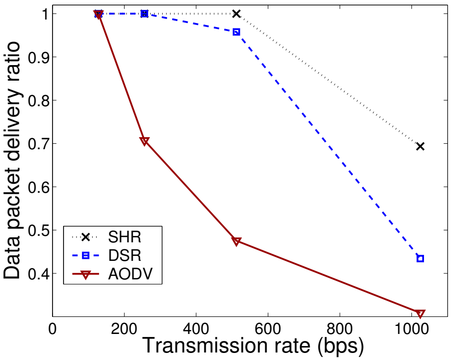


Fig. 7 The fraction of data packets successfully delivered as a function of transmission rate.

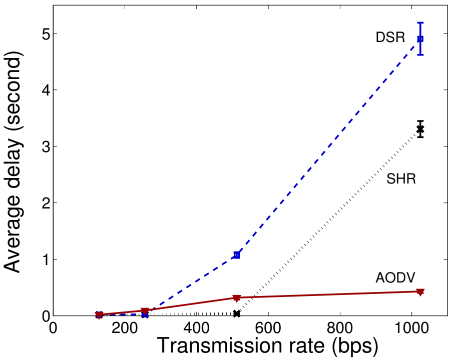
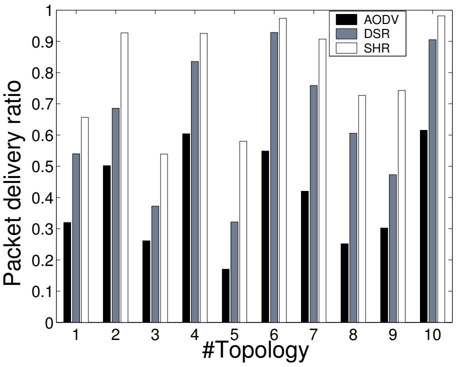
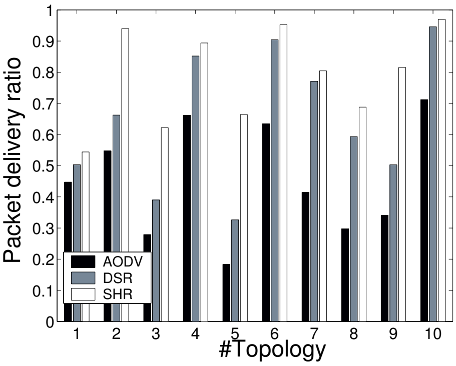


Fig. 8 Average delay as a function of transmission rate.



(a) 10 nodes join.



(b) 10 nodes leave.

Fig. 9 The fraction of data packets successfully delivered on ten randomly generated topologies in dynamic scenarios.

poor performance due to flooding of a large number of control packets. Therefore, we decide to randomly start each traffic source between $0^{th} - 50^{th}$ second of the simulation. Since there is only one destination (a base station), each source node can use discovered route for the entire simulation. Both approaches will discover a new route when link failure is detected.

3. 2 Simulation Results

The packet delivery ratio and average delay with 99% confidence interval are shown in **Figs. 7** and **8**, respectively. It is clear from the figures that all of the protocols deliver a greater percentage of the originated data packets at low traffic loads. In particular, the delivery ratio of three protocols is nearly 100% at 128-bps traffic load. When the transmission rate is increased to 256 bps, DSR and SHR can still deliver nearly 100%, while the performance of AODV dramatically degrades to 57%. When we increase the load to 512 bps, only SHR can still deliver nearly 100%, while the delivery ratio of DSR drops to 89% and AODV can deliver only 67%. At the extreme case in our simulation, i.e., 1024-bps load, the delivery ratio of SHR sharply drops to 70% which is still much better than one-third delivery ratio of DSR and AODV which is an unacceptable value for any applications. A main reason of dropped packets is no available route dut to high congestion and high collision. This is an effect of flooding which incurs a large number of control and data packets. When considering the average delay, SHR delivers slightly faster than DSR at both 128- and 256-bps load. However, SHR still does very well at 512-bps load, i.e., less than 0.05 second, while DSR delivers at an average of 3 seconds. As one would expect, SHR delivers faster than DSR about 2 seconds at 1024-bps load. AODV has lower delay than both SHR and DSR at the high load because there are many dropped packets especially packets that traverse over long distance. We note that such dropped packets are not included in delay calculation.

We also simulate two scenarios of dynamic networks, joining and leaving scenarios. For the joining scenario, 10 nodes are randomly deployed at 100 second in addition to 50 nodes deployed at the beginning of the simulation. These additional nodes need to find a parent for communicating. To simulate the leaving scenario, we deploy 60 nodes and apply an energy model in our simulation

---

ber of sensors are deployed and start to sense information at the same time.

by providing much enough energy for the entire simulations for 50 nodes and making the battery of 10 nodes depletes at some point of time before the simulations end. We show only the delivery ratio of 1024-bps load on ten random topologies which is appropriate for evaluating the resilience of protocol since it is an extreme case. Other parameters (simulation area, radio coverage, etc.) are same as the above simulations. The delivery ratio of joining and leaving scenarios are shown in **Figs. 9(a)** and **9(b)**, respectively. For the joining case, the delivery ratio of SHR is about 10-20% better than DSR which in turn is higher than AODV about 10-20%. When considering the leaving case, SHR still delivers greater percentage of the originated data packets than DSR. AODV is the worst amongst three protocols.

4. Related Work

DSDV 10), DSR 11), and AODV 12) are routing protocol developed for mobile ad hoc network which is very different from sensor network in an issue that the nodes are mobile. Since MANET is dynamic, the routing protocols must adapt to current physical topology by periodically updating the states for proactive protocol (DSDV) or using reactive approach (DSR and AODV). Each node in such protocols keeps the routing table for all destinations, in contrast, each node in SHR keeps only the knowledge of a parent node. Moreover, SHR does not use periodical updating as DSDV or floods the control packets as DSR and AODV.

LAR 13), GPSR 14), and GEAR 15) are geographic routing protocols that use location information to decrease the overhead of route discovery and find the routes quickly. Unfortunately, such approaches may not implement into many sensor networks because each sensor node requires to know its exact geographic location. Current methods of determining geographic location 23)∼25) consume much energy and may not be possible in many sensor network scenarios. Moreover, location-aware module increases the production cost for the sensor nodes, especially in large-scale sensor networks. It also wastes to use location-aware nodes in non-mobile networks. Note again that geographical information is not required for routing in SHR.

Directed diffusion 18) is a data-centric routing based on the name of data. Base stations draw interesting information by flooding the interests and setting up gradients within the network. It also provides in-network aggregation. Directed diffusion is a query-style protocol dealing with the name of data which is completely different from SHR. Therefore, directed diffusion is not appropriate for comparison with SHR.

Randomized algorithm is used in constrained random walk 26) to determine the next hop in order to achieve load balancing. However, they consider only one-source network in their evaluation because multiple-source network requires more complex computation to balance energy. In contrast, SHR considers the networks composed of multiple sources and sinks.

5. Conclusions

This paper has demonstrated that SHR protocol efficiently collects the data packets across multi-hop wireless sensor networks while maintaining a constant amount of local state to nearly stateless and making only local decisions. Since the state required on each node is very low and independent of both network size and network density, SHR is highly scalable. In particular, there is no need to maintain the state about all neighboring nodes and the interactions between nodes are strictly local. Consequently, the nodes can make quick decisions and the hierarchical tree is agilely constructed. SHR also supports dynamic properties introduced in this paper. In other words, it is a self-organized protocol according to the joining or leaving nodes. Furthermore, it avoids flooding and periodic updating that incurs high traffic load. Geographical information is also not required in our protocol. We have examined the efficiency of SHR in terms of data packet delivery ratio and average delay through the *ns*-2 simulator. The results have shown that SHR achieves notably high delivery ratio and low delay, and it is also tolerable to high traffic loads.

References

1) J. Hill and D. Culler: "Mica: a wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, (2002).

2) J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister: "System architecture directions for networked sensors," in *Proc. of the 9th international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Cambridge, MA, pp. 93–104, Nov. (2000).

3) K. Mechitov, W. Kim, G. Agha, and T. Nagayama: "High-frequency distributed sensing for structure monitoring," in *Proc. of the First International Workshop on Networked Sensing Systems (INSS)*, Tokyo, Japan, pp. 101–104, June (2004).

4) H. Wang, J. Elson, L. Girod, D. Estrin, and K. Yaot: "Target classification and localization in habitat monitoring," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, pp. 844–847, Apr. (2003).

5) A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson: "Wireless sensor networks for habitat monitoring," in *Proc. of the 1st ACM International Workshop*

on Wireless Sensor Networks and Applications (WSNA), Atlanta, GA, pp. 88–97, Sept. (2002).

6) A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao: "Habitat monitoring: application driver for wireless communications technology," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 2 supplement, pp. 20–41, (2001).
7) D. C. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole: "Research challenges in environmental observation and forecasting systems," in *Proc. of the 6th International Conference on Mobile Computing and Networking (MOBICOM)*, Boston, Massachusetts, pp. 292–299, Aug. (2000).
8) S. Kim, D. Culler, and J. Demmel: "Structural health monitoring using wireless sensor networks," (2003). http://www.eecs.berkeley.edu/∼binetude/course/cs294_1/paper.pdf.
9) A. S. Tanenbaum: *Computer Networks.* Prentice Hall PTR, 4 ed., Aug. (2002).
10) C. E. Perkins and P. Bhagwat: "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. of ACM SIGCOMM*, London, UK, pp. 234–244, Sept. (1994).
11) D. B. Johnson, D. A. Maltz, and J. Broch: "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," in *Ad Hoc Networking* (C. E. Perkins, ed.), ch. 5, pp. 139–172, Addison-Wesley, (2001).
12) C. E. Perkins, E. M. Belding-Royer, and S. Das: "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, IETF, July (2003).
13) Y.-B. Ko and N. H. Vaidya: "Location-aided routing (LAR) in mobile ad hoc networks," in *Proc. of the 4th International Conference on Mobile Computing and Networking (MOBICOM)*, Dallas, Texas, pp. 66–75, Oct. (1998).
14) B. Karp and H. T. Kung: "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. of the 6th International Conference on Mobile Computing and Networking (MOBICOM)*, Boston, Massachusetts, pp. 243–254, Aug. (2000).
15) Y. Yu, R. Govindan, and D. Estrin: "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," Technical Report UCLA/CSD-TR-01-0023, UCLA, May (2001).
16) H. Lundgren, E. Nordstro, and C. Tschudin: "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *Proc. of International Workshop on Wireless Mobile Multimedia (WoWMoM)*, Atlanta, GA, pp. 49–55, Sept. (2002).
17) S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler: "Supporting aggregate queries over ad-hoc wireless sensor networks," in *Proc. of the 4th IEEE Workshop on Mobile Computing and Systems Applications (WMCSA)*, Callicoon, NY), pp. 49–58, June (2002).
18) C. Intanagonwiwat, R. Govindan, and D. Estrin: "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proc. of the 6th International Conference on Mobile Computing and Networking (MOBICOM)*, Boston, Massachusetts, pp. 56–67, Aug. (2000).
19) D. Petrovic, R. C. Shah, K. Ramchandran, and J. M. Rabaey: "Data funneling: Routing with aggregation and compression for wireless sensor networks," in *Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, Anchorage, AK, pp. 156–162, May (2003).
20) "Network simulator – ns-2." http://www.isi.edu/nsnam/ns/.
21) J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva: "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. of the 4th International Conference on Mobile Computing and Networking (MOBICOM)*, Dallas, Texas, pp. 85–97, Oct. (1998).
22) "Mica2 mote." http://www.xbow.com/index.htm.
23) P. Bahl and V. N. Padmanabhan: "RADAR: an in-building rf-based user location and tracking system," in *Proc. of IEEE INFOCOM*, Tel-Aviv, Israel, pp. 775–784, Mar. (2000).
24) L. Doherty, K. S. J. Pister, and L. E. Ghaoui: "Convex optimization methods for sensor node position estimation," in *Proc. of IEEE INFOCOM*, Anchorage, Alaska, pp. 1655–1663, Apr. (2001).
25) N. B. Priyantha, A. Chakraborty, and H. Balakrishnan: "The cricket location-support system," in *Proc. of the 6th International Conference on Mobile Computing and Networking (MOBICOM)*, Boston, Massachusetts, pp. 32–43, Aug. (2000).
26) S. D. Servetto and G. Barrenechea: "Constrained random walks on random graphs: Routing algorithms for large scale wireless sensor networks," in *Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, Atlanta, Georgia, pp. 12–21, Sept. (2002).

Niwat THEPVILOJANAPONG

received his B.E. degree in Electrical Engineering from Chulalongkorn University (Thailand) in 1999, and his M.E. degree in Information and Communication Engineering from the University of Tokyo in 2003. He is currently pursuing the Ph.D. degree in Information and Communication Engineering at the University of Tokyo. His research interests include wireless sensor networks, mobile ad hoc networks, ubiquitous computing, and network measurement. He is a student member of the ACM, the IEEE, and the IEEE Communications Society.

Yoshito TOBE (Member)

received Ph.D. in Media and Governance from Keio University in 2000. He is currently a Professor in the Department of Information Systems and Multimedia Design at Tokyo Denki University. His research includes multimedia communications, Internet measurement, wireless sensor networks, and ubiquitous networked sensing. He is a member of ACM, IEEE Communications Society, SICE, and IPSJ.

Kaoru Sezaki

received his B.E., M.E. and Dr.E. degrees in electrical engineering from the Univ. of Tokyo in 1984, 1986, and 1989, respectively. He joined the Institute of Industrial Science at the Univ. of Tokyo in 1989. He is now an associate professor at the Center for Spatial Information Science at the University of Tokyo. From 1994 to 2000, he was a guest associate professor at the National Center for Science Information Systems. He has been a guest associate professor at National Institute of Informatics since 2000. he has been a special member of Telecommunications Business Dispute Settlement Commission since 2001. From 1996 to 1997, he was a visiting scholar at the Univ. of California, San Diego. His research interests include communication networks, location- and context-aware network services, high-speed switching systems, collaboration systems with haptics, GIS and image processing. He is a member of the IEEE.

Reprinted from Trans. of the SICE
Vol. E-S-1 No. 1 43/51 2006